

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION  
International Bureau



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<b>(51) International Patent Classification <sup>6</sup> :</b> <b>G07F 7/10</b>		<b>A2</b>	<b>(11) International Publication Number:</b> <b>WO 98/52160</b>
			<b>(43) International Publication Date:</b> 19 November 1998 (19.11.98)
<b>(21) International Application Number:</b> PCT/GB98/01393 <b>(22) International Filing Date:</b> 14 May 1998 (14.05.98)  <b>(30) Priority Data:</b> 60/046,514 15 May 1997 (15.05.97) US 09/076,550 12 May 1998 (12.05.98) US  <b>(71) Applicant:</b> MONDEX INTERNATIONAL LIMITED [GB/GB]; 47-53 Cannon Street, London EC4M 5SQ (GB).  <b>(72) Inventors:</b> EVERETT, David, Barrington; 31 Ashdown Avenue, Saltdean, Brighton, East Sussex BN2 8AH (GB). MILLER, Stuart, James; 9 Woodford Green, The Warren, Bracknell, Berkshire RG12 9YQ (GB). PEACHAM, Anthony, David; 4 Lynwood, Groombridge, Tunbridge Wells, Kent TN3 9LX (GB). SIMMONS, Ian, Stephen; The Elms, School Road, Broughton, Cambridgeshire PE17 3AT (GB). RICHARDS, Timothy, Philip; 32 Graig Mount, Radlett, Hertfordshire WD7 7LW (GB). VINER, John, Charles; Hydes, Woodlands Lane, Windlesham (GB).  <b>(74) Agent:</b> POTTER, Julian, Mark; D. Young & Co., 21 New Fetter Lane, London EC4A 1DA (GB).			<b>(81) Designated States:</b> AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).  <b>Published</b> <i>Without international search report and to be republished upon receipt of that report.</i>
<b>(54) Title:</b> SYSTEM AND METHOD FOR FLEXIBLY LOADING AN IC CARD			
<b>(57) Abstract</b> <p>A system and method of flexibly loading an application and its associated data from an application provider onto an IC card. The application and its associated data is divided into segments which can each fit into the input buffer of an Integrated circuit card. Each segment is transmitted separately and the Integrated circuit card then stores the segment in an available space in the IC card's memory. The segments can be placed in non-contiguous memory in order to reduce memory fragmentation.</p>			
<p>The diagram illustrates the memory layout of an IC card (300). It consists of a vertical stack of memory blocks. At the top is 'OPERATING SYSTEM DATA 305' (351). Below it is 'SEGMENT 307' (353). Then is 'APPLICATION 301' (355), which is grouped with 'SEGMENT 309' (357) by a bracket labeled '313'. Below that is 'APPLICATION 303' (359), which is grouped with 'SEGMENT 311' (361) by a bracket labeled '300'. At the bottom is an empty space (363).</p>			

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

SYSTEM AND METHOD FOR FLEXIBLY LOADING AN IC CARD

## BACKGROUND OF INVENTION

Integrated circuit ("IC") cards are becoming increasingly used for many  
5 different purposes in the world today. An IC card (also called a smart card) typically is  
the size of a conventional credit card which contains a computer chip including a  
microprocessor, read-only-memory (ROM), electrically erasable programmable read-  
only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to  
support the microprocessor in its operations. An IC card may contain a single application  
10 or may contain multiple independent applications in its memory. MULTOS™ is a  
multiple application operating system which runs on IC cards, among other platforms,  
and allows multiple applications to be executed on the card itself. This allows a card user  
to run many programs stored in the card (for example, credit/debit, electronic  
money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM,  
15 telephone and/or POS) in which the card is inserted for use.

A conventional single application IC card, such as a telephone card or an  
electronic cash card, is loaded with a single application when it is manufactured and  
before it is given to a card user. That application, however, cannot be modified or  
changed after the card is issued even if the modification is desired by the card user or card  
20 issuer. Moreover, if a card user wanted a variety of application functions to be performed  
by IC cards issued to him or her, such as both an electronic purse and a credit/debit  
function, the card user would be required to carry multiple physical cards on his or her

person, which would be quite cumbersome and inconvenient. If an application developer or card user desired two different applications to interact or exchange data with each other, such as a purse application interacting with a frequent flyer loyalty application, the card user would be forced to swap multiple cards in and out of the card-receiving  
5 terminal, making the transaction difficult, lengthy and inconvenient.

Therefore, it is beneficial to store multiple applications on the same IC card. For example, a card user may have both a purse application and a credit/debit application on the same card so that the user could select which type of payment (by electronic cash or credit card) to use to make a purchase. Multiple applications could be  
10 provided to an IC card if sufficient memory exists and an operating system capable of supporting multiple applications is present on the card. Although multiple applications could be preselected and placed in the memory of the card during its production stage, it would also be beneficial to have the ability to load and delete applications for the card post-production as needed.

15 It is important, particularly where there is a continuing wide availability of new applications to the cardholder, that the system has the capability of adding applications onto the IC card subsequent to issuance. This is necessary to protect the longevity of the IC cards; otherwise, once an application becomes outdated, the card would be useless. It would be beneficial to allow the addition of applications from a  
20 remote location as well as from a direct connection to an application provider's terminal.

For example, it would be beneficial for a card user to be able to plug his IC card into his home computer and download an application over the Internet. Alternatively, it would be beneficial for an application provided by Bank A to be loaded from a terminal (such as an ATM) located at Bank B which is connected to Bank A by a network or series of  
5 interconnected networks.

The increased flexibility and power of storing multiple applications on a single card create new technical challenges to be overcome concerning the application loading process in which information (including application code and associated data) is exchanged between the application provider and the individual card. The IC card only  
10 has a finite amount of memory on the card for storing applications. Applications and their associated data can vary drastically in size depending upon the application. When multiple applications are stored on a card, and a series of application additions and deletions have occurred, memory fragmentation where memory which is free cannot be used because of size limitations.

15 Additionally, an IC card has limited space in its input buffer, which can be separate or combined with an output buffer, i.e., an Input/Output (I/O) buffer. It may not be possible to fit the entire application and its associated data into an I/O buffer of an IC card at one time. In order to achieve the flexibility of selectively loading and deleting applications on an IC card, the problems of limited I/O buffer space and fragmentation  
20 must be addressed.

Accordingly, it is an object of preferred embodiments of this invention to provide a system and method that allows for flexible loading of an application and its associated data onto an IC card by segmenting the application and associated data into selected segments in order to limit the size of the data packets being transmitted at one  
5 time and reduce fragmentation in the memory of the IC card.

### SUMMARY OF THE INVENTION

10 These and other objectives are achieved by an embodiment of the present invention which provides an IC card system and method for flexibly loading an application and its associated data from an application onto an IC card. The application provider divides the application and its associated data into segments which will fit into the I/O buffer of the intended IC card. Each segment is transmitted separately and the IC  
15 card stores the segment in an available space in the IC card's memory. The segments can be placed in non-contiguous memory in order to reduce memory fragmentation. The IC card's microprocessor can additionally determine the smallest memory space which will store the segment in order to minimize fragmentation.

In a preferred embodiment, the application provider determines the size of the IC  
20 card's I/O buffer so that it can correctly select the size of each segment.

### BRIEF DESCRIPTION OF THE DRAWINGS

5                   Further objects, features and advantages of embodiments of the invention will become apparent from the following detailed description taken by way of example only and in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

                  Fig. 1 is block diagram of the flexible loading system of the present  
10   invention;

                  Fig. 2 is a block diagram of an IC card chip upon which an application and its associated data can be flexibly loaded and stored;

                  Fig. 3 is a graphic example of a memory map of EEPROM on an IC card;

                  Fig. 4 is a flow chart of an example of multiple segments being loaded  
15   onto the IC card;

                  Fig. 5 is a flow chart of the steps of segmenting the application and its associated data by the application provider; and

                  Fig. 6 is a flow chart of the steps of receiving and processing the segmented information by the IC card.

20                  Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now



be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

5

### DETAILED DESCRIPTION OF THE INVENTION

It is beneficial to have the capability to load applications onto IC cards  
10 containing multiple application operating systems at any time during the lifetime of the IC card. This flexibility allows a user of a card to periodically add new applications to the IC card and also allows older applications to be updated with newer versions of the application when they are released. For example, a card user may start with an IC card that contains a purse, or electronic cash application (e.g., MONDEX™), being stored on  
15 her IC card. Some time after the user has the card, she may load an additional application onto the card such as a credit/debit application. Some time after loading the credit/debit application on the card, a new version of the credit/debit application may become available and the card user should be able to erase the old application on her IC card and replace it with the new version of the credit/debit application which may contain  
20 additional features.

In a multiple application operating system environment, it is beneficial to be able to load applications both at terminals, such as a bank ATM machine, as well as

over remote communication links, such as telephone lines, cable lines, the Internet, satellite or other communications means. When loading applications onto an IC card, the application provider and the card issuer (which could be the same entity) need to provide security regarding the applications to be loaded. The IC card has a limited amount of available I/O space and a limited amount of memory to store applications and their associated data. In order to address these limitations, the application and its associated data is segmented such that each segment will fit in the IC card's I/O buffer at one time. The segment is then stored in the IC card's storage memory, e.g., EEPROM, in a manner that can be non-contiguous to reduce memory fragmentation. This system and technique will now be described in detail.

Figure 1 shows a block diagram of the entities used in a remote application loading process of an application and its associated data. While Figure 1 shows a remote loading system, the flexible loading technique also applies to local loading such as a terminal located at the application provider. System 100 includes an application provider for transmitting an IC card application and its associated data to an intended IC card 103, an IC card interface device 105 and a data conduit 107. The application provider 101 can be a card issuer, bank or other entity which provides application loading services. The application provider 101 preferably initiates an application loading process onto IC card 103. Alternatively, the IC card 103 can request the loading process. Application Provider 101 is connected to data conduit 107 which is connected to interface device 105

(e.g., a terminal that communicates with an IC card). Data conduit 107 can be a telephone line, an intranet, the Internet, a satellite link or any other type of communications link. The application provider 101, which is remotely located from the IC card 103 in this example, desires to send and load an application to the IC card.

5 Application provider 101 has an I/O buffer 113 and IC card 103 has an I/O buffer 115. In addition, interface device 105 also contains an I/O buffer 117. Each of the I/O buffers has a maximum storage capacity. The I/O buffers could be a combined input or output buffer or the input buffer and output buffer could be separate. However, the IC card 103 will typically have the smallest I/O buffer due to physical size limitations. The IC card  
10 103 also has a memory 119 in which it stores the loaded application and its associated data.

In the illustrative embodiment of Figure 1, the application provider 101 sends two application segments S1, 109 and S2, 111 to the interface device 105 which is coupled to IC card 103. The application segments are discussed in more detail in  
15 connection with Figure 4. The application and its associated data are broken into two or more segment units in order for each of the data segments to fit in the I/O buffer of the I/O card. Additionally, the segmentation of the application and associated data helps to reduce fragmentation of the memory of the IC card which stores the application and associated data being loaded.

Figure 1 shows two segments 109 and 111 which are transferred at discrete times from the application provider to the IC card. However, any number of segments could be used depending upon such factors including the size of the application being loaded, the size of the associated data being loaded, the size of the respective I/O buffers, the availability of memory space on the IC card and the amount of memory fragmentation already on the IC card.

The application could be loaded directly at a terminal and not remotely. In that case, a separate interface device 105 would not be required because the application provider would have its own terminal capable of communicating with the IC card. For example, a bank could load an application onto an IC card by requiring the customer to insert his or her card into the bank's ATM machine. In that case, the application provider communicates with the IC card locally and transmissions are not sent over telephone lines or the Internet. Embodiments of the present invention are applicable to both the remote loading and local loading.

Figure 2 shows an example of a block diagram of an IC card chip upon which an application can be flexibly loaded and stored. An integrated circuit is located on an IC card for use. The IC card preferably includes a central processing unit 201, a RAM 203, an EEPROM 205, a ROM 207, a timer 209, control logic unit 211, an I/O port 213 and security circuitry 215, which are connected together by a conventional data bus.

Control logic 211 in memory cards provides sufficient sequencing and switching to handle read-write access to the card's memory through the input/output ports. CPU 201 with its control logic can perform calculations, access memory locations, modify memory contents, and manage input/output ports. Some cards have a coprocessor  
5 for handling complex computations like performing cryptographic operations.

Input/output ports 213 are used under the control of a CPU and control logic, for communications between the card and a card interface device. Input/Output ports 213 include an I/O buffer. Timer 209 (which generates or provides a clock pulse) drives the control logic 211 and CPU 201 through the sequence of steps that accomplish memory  
10 access, memory reading or writing, processing, and data communication. A timer may be used to provide application features such as call duration. Security circuitry 215 preferably includes fusible links that connect the input/output lines to internal circuitry as required for testing during manufacture, but which are destroyed ("blown") upon completion of testing to prevent later access. The application segments are stored in  
15 EEPROM 205. The storage and memory management process as described herein is performed by the CPU 201.

Figure 2 also shows a possible configuration for the integrated circuit for the application provider. CPU 201 present in the integrated circuit for the application provider determines the size of the IC card's I/O buffer, controls the segmentation of the

application and associated data described herein and performs any other necessary operation.

Figure 3 shows a graphic representation of a memory map of EEPROM 300 on IC card 103. In this illustrative example, three applications are stored in EEPROM of an IC card. The first application 301 is stored in a contiguous memory space 355. Contiguous memory space means that the application occupies sequential memory addresses with no skipped memory addresses. A second application 303 is stored in contiguous memory space 359. Operating system data required for the execution of the operating system is stored in memory space 351. One example of a cause of fragmentation existing in the IC card is a previous application being deleted which was previously located at memory space 313. The next application loaded onto the IC card after the initial application was deleted can be a different size than the initial application and thus not all the freed up available memory space can be used in such a manner where two or more programs and data are stored contiguously without leaving small portions of unused memory space. In the example of Figure 1, the last application and its associated data which was loaded was segmented into three segments 307, 309 and 311. These segments are smaller portions of the entire application and its associated data set which could be placed in smaller areas of available memory. Thus fragmentation in the IC card's memory was alleviated by segmenting the application and its associated data.

The operating system stored on the card maintains a record of the physical location of the different segments and can access the physical locations when a logical address is called out when a program or operating system is being executed. The physical address look-up data can be stored in a table, a stack, a pointer or any conventional means for indicating the physical locations. Memory space 363 in Figure 3 is shown as not storing any data in the example and that memory space could be later used for storing new segmented applications and their associated data.

Figure 4 shows a flow chart of an illustrative example of loading multiple segments into a memory of an IC card from an application provider. In this example, six initial segments are created to be loaded onto the IC card. Two of the segments are further divided into components which results in a total of nine segments individually being sent to the IC card.

Step 401 loads a segment corresponding to the program code of the application to be provided to the IC card. The program code includes the program instructions which will be executed by the microprocessor located on the IC card. If the code segment is too large to fit into the I/O buffer of either the IC card or the application provider, then the segment can be further split into two or more components which can be separately transmitted to the IC card. In Figure 4, three components are illustrated for the program code, components 413, 415 and 417. The components are preferably stored in contiguous memory locations in the memory of the IC card. However, the components

can be stored in non-contiguous locations if component pointers or tables are supported by the operating system on the IC card.

Step 403 loads the application data segment onto the IC card. The application data segment includes necessary and optional data needed for the execution of the application code. For example, if the application is a credit/debit application, the card user's account number, identification data and credit limit may be needed for the application to run. Another example is a health related application where a customer's medical history may be stored on the card for quick access at remote locations. The medical history data may be quite large and require further segmentation into two or more components. In Figure 4, components 419 and 421 are shown as subsets of the data segment being loaded in step 403.

Step 405 loads a Key Transformation Unit (KTU) segment for the application being loaded. If the application is being loaded from a remote location, there is a need to make sure the transmission is secure from third party access. The KTU information preferably contains information regarding the encryption key used to encipher the application program and associated data. The key information is sent with the application because applications can be transmitted from any application provider to any IC card with an IC card system. Since different encryption techniques can be used by different application providers, the KTU information is necessary. However, the flexible



loading technique also applies when no encryption scheme is used and this information could also be included in another segment depending upon its size.

Step 407 loads a file control segment onto the IC card. File control information preferably includes an application identifier, security information and application and data size requirements. The file control information will be used by the operating system on the IC card to process the application. While in this example the file control information is a separate segment, it could be included in another segment depending upon its size.

Step 409 loads a directory information segment onto the IC card. The directory information preferably includes the name of the application which can be used by the operating system to identify the application. For example, if a select file command is initiated by a terminal, the name of the file to be selected which accompanies the command will be recognized by the operating system on the IC card. If the MONDEX™ Purse is selected by a customer as a terminal, the terminal will send a command to the IC card in the form of a "Select File Mondex" and the IC card will correlate MONDEX with a previously loaded application with the directory name Mondex. While in this example the directory information is a separate segment, the information could be included in another segment depending upon its size.

Step 411 loads an application signature segment onto the IC card. The application signature segment preferably includes data signed with the digital signature of

the application provider. This allows the IC card to verify that the application provider is the genuine application provider and not an imposter. The IC card verifies the signature with the public key of an asymmetric encryption key pair of the application provider.

While in this example the application signature is a separate segment, the information  
5 could be included in another segment if its size permitted it.

The segments could be organized in any manner and sent in any order. The IC card will need to have identified the subject matter of the incoming segment or component so that it can later locate a specific segment or component when needed. This information can be part of the load control information or can be obtained prior to the  
10 loading of the application. While Figure 4 describes a number of different segments, the subject matter of the segments transmitted will vary and depend upon the particular application and associated data.

Figure 5 shows a flowchart of the steps the application provider performs when segmenting the application and associated data to be loaded upon the IC card. Step  
15 501 determines the I/O buffer size of the IC card. Alternatively, the input buffer size is determined if the input and output buffers are separate on the IC card. In most cases, the IC card I/O buffer will be smaller than the application provider I/O buffer because of the limited memory on the IC card. However, if the application provider I/O buffer or the Interface I/O buffer is smaller than the IC card I/O buffer, the smallest I/O buffer will  
20 control the size of the segments. The application provider can determine the IC card

memory buffer size by some preliminary information exchange which identifies the IC card as the correct card upon which to load the application. Alternatively, some agreement or standard can be followed so that the application provider can create segments which will fit in an IC card which follows the agreement or standard.

5                   Step 503 then segments the application and associated data in two or more segments. In the example of Figure 4, six initial segments were created and some of the segments were further divided to form two or more components. The segmented information is preferably divided in a predetermined organization to aid the IC card processing of the segments.

10                   Step 505 then sends the segments to the IC card one at a time. When the IC card receives a segment in its I/O buffer, it will store that segment in a location of its memory thus freeing up its I/O buffer for the next incoming segment. After all the segments have been transmitted, the application provider can send a transmission indicating no more segments are being transmitted or the number of segments can be sent  
15 at the beginning of the transmission. Alternatively, a known segment protocol can be followed.

                  Figure 6 is a flow chart of the steps of processing the segmented information performed by the IC card. Step 601 receives a transmitted segment in the I/O buffer of the IC card. The entire segment will fit within the I/O buffer because of the  
20 processing performed at the application provider. Step 603 then stores the segment in

available memory space after the microprocessor on the IC card identifies the proper memory space. The processor can check for the first available free memory space that is sufficient to store the segment. Once the segment is stored at a physical location, that location is recorded either in a segment address table, by a pointer or by any other

5 conventional means. Different memory architectures can be used for storing the segments. For example, all the similar types of segments (e.g., program code) for the stored applications can be stored contiguously if desired. Alternatively, the processor can determine the space that is closest in size to the segment to be stored by scanning the memory. This will reduce any problems of fragmentation in the limited size IC card

10 memory.

Step 605 determines if there are any additional segments to be stored. This step can be accomplished by checking earlier information regarding the number of segments which were being sent. It can also be accomplished by receiving a transmission indicating no more segments. Alternatively, the IC card can simply remain in a wait

15 status until additional data or instructions is sent to the card. If the IC card determines that additional segments are being transmitted, the technique jumps back to step 602. If no more segments, the process ends.

The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems

and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.

For example, while loading an application and its associated data is discussed herein, the same flexible loading process can apply to transmitting other types  
5 of data such as data blocks, database files, word processing documents or any other type of data requiring to be transmitted in a segmented manner.

The scope of the present disclosure includes any novel feature or combination of features disclosed therein either explicitly or implicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or  
10 mitigates any or all of the problems addressed by the present invention. The application hereby gives notice that new claims may be formulated to such features during the prosecution of this application or of any such further application derived therefrom. In particular, with reference to the appended claims, features from dependant claims may be combined with those of the independent claims in any appropriate manner and not merely  
15 in the specific combinations enumerated in the claims.

WE CLAIM:

1 1. A method for loading an application and its associated data from an  
2 application provider onto an integrated circuit card, wherein said integrated  
3 circuit card comprises a memory, comprising the steps of:  
4 dividing said application and its associated data into a  
5 plurality of segments;  
6 separately transmitting each said segment to said  
7 integrated circuit card; and  
8 storing each said separately transmitted segment in an  
9 available area of said integrated circuit card's memory.

1 2. The method of claim 1, wherein at least two of said plurality of  
2 segments are not stored contiguously.

1 3. The method of claim 1 or claim 2, further including the step of  
2 determining an available area in said integrated circuit card's memory to store  
3 each said segment.

1 4. The method of claim 3, wherein said determining step identifies the  
2 smallest available area in said integrated circuit card's memory in which said  
3 segment can be stored.

1 5. The method of any preceding claim, wherein at least a first portion of  
2 said application is not stored contiguously with said application's remaining  
3 portion in said integrated circuit card's memory.

1 6. The method of any preceding claim, wherein said application is not  
2 stored contiguously with said associated data in said integrated circuit card's  
3 memory.

1 7. The method of any preceding claim, wherein said application is  
2 divided into a plurality of segments.

1 8. The method of any preceding claim, wherein said associated data is  
2 divided into a plurality of segments.

1 9. The method of any preceding claim, further including the step of  
2 determining said integrated circuit card's input buffer size.

1 10. A system for loading an application and its associated data onto an  
2 integrated circuit card comprising:  
3 an application provider comprising means for dividing  
4 said application and its associated data into a plurality of segments and means  
5 for separately transmitting each said segment to said integrated circuit card;  
6 and  
7 an integrated circuit card comprising a memory, means  
8 for receiving said transmitted segments and means for storing each said  
9 transmitted segment in an available area of said integrated circuit card's  
10 memory.

1 11. The system of claim 10, wherein at least two of said plurality of  
2 segments are not stored contiguously in said integrated circuit card.

1 12. The system of claim 10 or claim 11, wherein said card further  
2 includes means for determining an available area in said memory to store  
3 each said segment.



1 13. The system of claim 12, wherein said determining means identifies  
2 the smallest available area in which said segment can be stored.

1 14. The system of any of claims 10 to 13, wherein at least a first portion  
2 of said application is not stored contiguously with said application's  
3 remaining portion in said memory.

1 15. The system of any of claims 10 to 14, wherein said application is not  
2 stored contiguously with said associated data in said memory.

1 16. The system of any of claims 10 to 15, wherein said application is  
2 divided into a plurality of segments.

1 17. The system of any of claims 10 to 16, wherein said associated data is  
2 divided into a plurality of segments.

1 18. The system of any of claims 10 to 17, wherein said application  
2 provider further includes means for determining said integrated circuit card's  
3 input buffer size.

- 1 19. The system of any of claims 10 to 18, wherein said means for  
2 receiving said transmitted segments has a size capacity smaller than said  
3 application and associated data's size.
- 1 20. The system of any of claims 10 to 19, wherein said integrated circuit  
2 card is remotely located from said application provider.

1/4

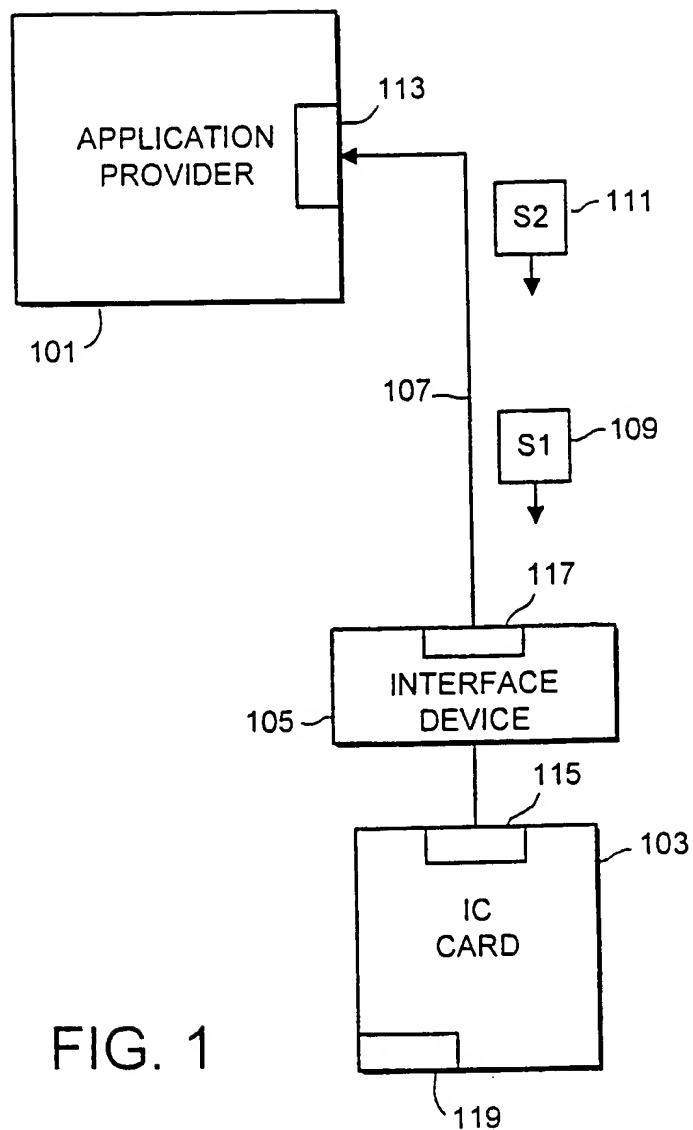


FIG. 1

100

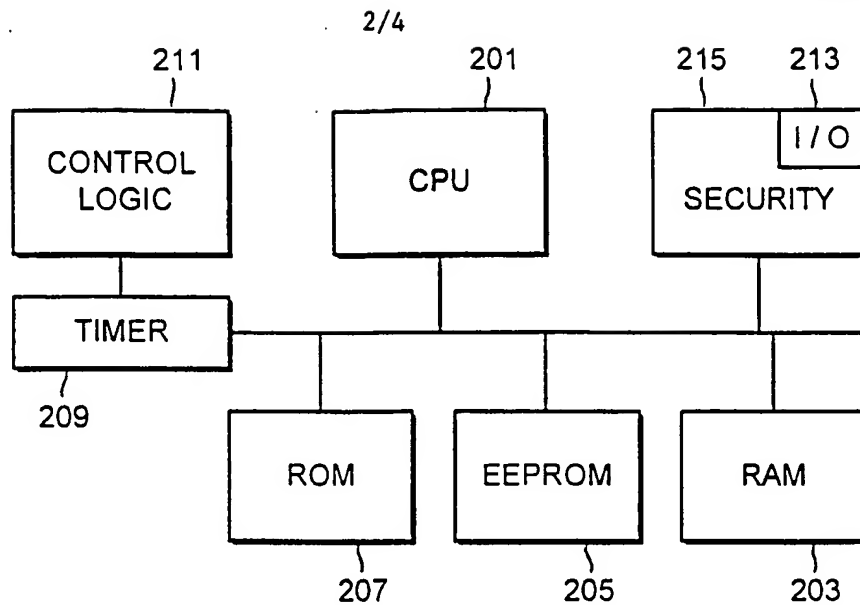


FIG. 2

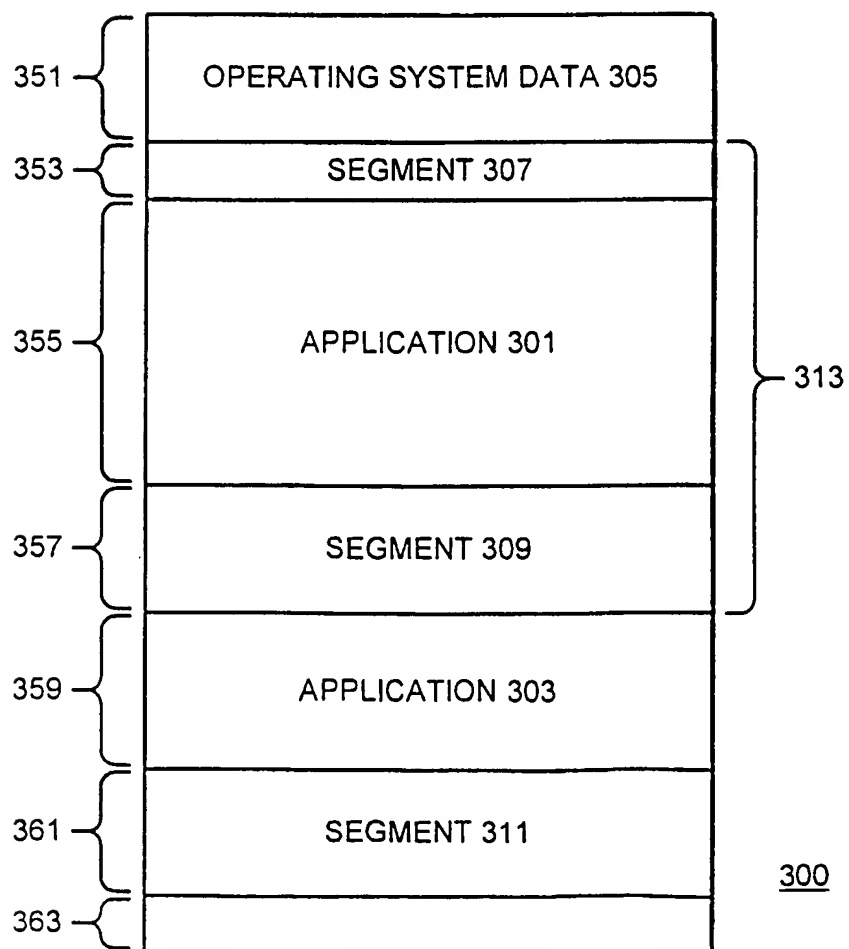


FIG. 3

3/4

